

# QUANTA

VOLUME 2 ISSUE 8

SEPTEMBER 1985

## Contents

2. Information on the Group
3. Editorial
3. Member's letters
7. Problem section
9. Making space on the arithmetic stack
9. Microdeal games review
10. The name table (entry for a parameter)
11. Translations and the Juki 6100
12. A BASIC extension function bug
12. Library corner
13. Statistics
14. Star Trek
15. Some Archive programming approaches
19. Using Quill
19. Loading D-Day
20. "AT" keyword co-ordinates
20. Installing the Epson MX-80 printer
21. Microdrive cartridge prices
21. Easel 2.00 tips
22. Extended precision arithmetic
23. D-Day by Games Workshop
25. Sending printer control codes in Quill etc.
26. BASIC programming hints
30. Stolen!
30. Disk corruption
31. Review of Talent's "Cartridge Doctor"
32. An additional note on Microdrive reliability
32. Backing-up Hisoft MONQL
33. Saving structures in C

### INFORMATION ON THE GROUP

Membership of QUANTA (QL, Users and Tinkerers Association) is by subscription to QUANTA, the group's newsletter, which is published monthly. Membership details are obtainable from the secretary. Membership of the group is open to anyone with an interest in the Sinclair QL microcomputer.

Members requiring assistance with problems related to the QL may write to or 'phone the secretary. An attempt will be made to put them in touch with a member who can help with the problem.

Workshops will be arranged from time to time in various parts of the country. The group maintains a software library. Most of the programmes are free to members. A library list will be published from time to time. Programmes are obtainable from the sub-librarians.

A list of members in any particular area is obtainable from the secretary. Please send all contributions to the newsletter to the editor.

Chairman and  
Newsletter Editor

Leon Heller,  
8, Morris Walk,  
Newport Pagnell,  
Bucks. MK16 8QD.  
Tel: (09081 613004

Secretary and  
Newsletter Publisher

Brian Pain,  
24, Oxford Street,  
Stony Stratford,  
Milton Keynes MK11 1JU  
Tel: (0908) 564271

Sinclair, QL, QDOS, QLUB and ZX Microdrive are trade marks of Sinclair Research Ltd.

Quill, Abacus, Archive and Easel are trade marks of Psion Ltd.

### NOTES FOR CONTRIBUTORS TO QUANTA

Quill files on Microdrive cartridge or 40 track disk (single or double sided), are preferred. Cartridges and disks will be returned ASAP.

Printed contributions are acceptable. They should be printed on A4 paper, using a fresh ribbon. Do not fold the sheets, but keep them flat when posting. Short letters and articles may be handwritten, but please write legibly.

Short programmes can be included, but lengthy programmes should be placed in the library, and a description sent in for the newsletter.

### EDITORIAL

By now, I suppose most of you will have heard that the price of the QL has been slashed to £199. Moreover, I've heard rumours that a new version with 512K of RAM and a built-in disk drive is likely, to compete with the latest Amstrad systems.

BBC News night recently sent a film crew and a reporter along to Milton Keynes, for the Sinclair feature that went out on Wednesday 21 August. We laid on a simulated local group meeting for them, and each of us was interviewed. The whole process took up most of a morning. Unfortunately, they didn't use any of it in the end! I thought the piece was quite fair to Sinclair, by the way.

I am starting to put together the programme for the Milton Keynes workshop, on the 8/9/10 November. Would any members who have any suggestions for lecture topics, or who are willing to give a lecture, please get in touch with me. Members who give lectures get a sizeable discount. Topics I think might be of interest are accounts programmes (like the new Sage package), Archive applications, programming languages, hardware add-ons.

Leon Heller

### MEMBERS' LETTERS

I have now found the bug in the machine code extension WHERE published in the May edition of QUANTA. The problem is that the utility CA\_CTINT corrupts register D6. If register D7 is used instead it should work correctly, so replace every D6 with D7. I have no excuse, it is documented. However, in the AH ROM it evidently doesn't happen.

My apologies to anyone who has been struggling to get the code to work for themselves. The programme and the Turtle graphics programmes which use it are now available from the library.

D. L. Coles,  
2, Boothey Close,  
Biggleswade,  
Beds.

-----

From an advert. in QL World, I ordered from \* \* Ltd. - 0 \* \* and \*  
on the 10 July 1985. The advert. stated immediate delivery but after five weeks  
I had not received any communication despite the fact that my cheque was banked  
by \* promptly. I wrote on the 14 August and requested the return of my money.

Seven weeks have now elapsed and I have still not had any communication  
from \* Can you give any advice as to the next step to take? \* BCP 4/9

Is it not possible to list firms that extract money with no service, and print the list in QUANTA to warn members not to deal with firms not prepared to honour agreements?

J. Willey,  
6, Lycium Close,  
Middlesborough,  
Cleveland TS7 8RS.

/\* I can only suggest that you pursue a claim through the small claims court. You shouldn't have any trouble getting your money back, and it won't cost you anything.

I'm not very keen on publishing a list of firms as you suggest, because circumstances can change, but I will publish letters such as yours, together with an explanation from the firm concerned, if possible. \*/

-----

I recently received a new copy of the QL USER GUIDE from Sinclair, dated 12/84. There is now an index in the KEYWORDS CONCEPTS section and also there are a lot less errata! The guide now mentions "ELLIPSE" & also "PRINT TO" and there are various other changes. I don't know if all other users have been sent this guide or if Sinclair have sent it to me in response to my many letters of complaint concerning the original guide.

My VER\$ 'AH' QL's 1st birthday passed recently and looking back over the past year I must say that overall MY QL has proved itself very reliable. I use my QL on average about 20 hours per week and it has never crashed for an inexplicable reason i.e. software faults in version 1 of the PSION software. Also I now have over 50 Microdrive cartridges none of which have ever failed and I have never lost a file from a Microdrive error. I can't complain about their speed either, Quill version 2.00 LOADs in under 15 seconds.

I have a suggestion to make concerning the format of contributions to QUANTA, Leon. Regarding your article 'Putting QUANTA together' on page 24 of QUANTA June '85, why not include your ideal QUILL parameters (i.e. Justification, page size etc.) as a blank '\_doc' file on one of the Library cartridges also you could include a suitable 'printer\_dat' file for use when 'printing' the finished letter to Microdrive file.

Kevin Leah,  
58, Mansfield Drive,  
Merstham,  
Redhill,  
SURREY RH1 3JN

/\* Since I reformat everything sent in for QUANTA, and print it out with Wordstar, it doesn't matter how it comes in. \*/

May I be permitted to make some comments on Benchmarks following John Tanner's letter in QUANTA 2.6.

When I first received my QL I was very keen, after all the bad press reviews, to see how it performed and I wrote a programme incorporating all the PCW (Personal Computing World) benchmarks, plus a few others that I came across. This programme is available to all on Library cartridge No. 3.

To make timing easier I used the Date function and, since this only measures to the nearest second, multiple runs and averaging were necessary for the shorter benchmarks. I now realise that more accurate timing is possible using the counter at memory 163886 which is incremented every 50th of a second, as recommended by Richard Kingslake in QUANTA 2.1.

```
100 poke_w 163886,0 :rem set counter to 0 lines of code
200 print peek_w(163886)*.02;" secs."
```

will time how long the programme takes between lines 100 & 200 to the nearest .02 of a second. The word at 163886 can of course accommodate only 32676 increments before being reset and so is useful only for timings of the order of minutes, not hours. It also seems to be disabled when the screen is written to and so gives low timings. It is however, suitable for the PCW Benchmarks.

I too have noticed that everyone quotes slightly different benchmarks. I think in part this is due to slightly different programming i.e. John quotes a benchmark on page 41 with the line

```
190 IF k<1000 then 140
```

Such a line is impossible on the QL but do you replace it with

```
190 IF k<1000 THEN GO TO 140
```

or by

```
190 IF k<1000:GO TO 140
```

The latter is slightly faster. Also there is a noticeable difference between  
100 PRINT "Done"!!DATE and 100 PRINT DATE! "Done"

with the latter being approx .03 sec faster. If timing is done by hand even greater differences are possible!

The major differences that John noticed with the values quoted in PCW, however are real. Sinclair really did improve the QL after the FB ROM which was the one which got reviewed and which is, in the main, the one which is still quoted. YOUR

COMPUTER of July 1984 quotes both sets of figures in a review by Kathleen Peel.

Like John, I also experimented with rewriting the PCW benchmarks in SuperBASIC and I summarise my findings thus.

FOR..... NEXT is slightly faster than FOR... ENDFOR

REPeat LOOP.....END REPeat LOOP is also slightly faster than the unstructured GOTO loop of the type

```
100 k=k-1
....code .....
200 IF k<1000:GO TO 100
```

However Procedure calls are slower than GOSUBs, adding about one second to the average benchmark. None of this accounts for John's poor results with the fully structured, multiple statement line versions.

I found that the real villain is putting a statement that initiates a loop of any kind (REPeat., FOR ...NEXT, GO SUB or even a Procedure call) anywhere but the beginning of a line. e.g. I find the benchmark that John lists as BM6 takes 24.06 secs. Place the GO SUB on line 150 after the  $a=K/2^3+4-5$  and the time goes up to 24.78 secs (Procedures give similar results but take slightly longer). Add the FOR L=1 to 5 onto the end and the time goes up to 30.38 secs. In the last case it doesn't much matter whether you use a the short or long form of the FOR... NEXT loop. I would guess that on return from any sort of loop, control passes to the start of the line the loop originated on and if the loop actually starts at a later statement it takes time to work its way to the correct statement.

Anyway, the message is clear. Unless you are trying to waste time don't begin any loop except at the start of a line.

I discovered some other unusual 'features'. A line of the type

```
190 k=0
200 REPeat LOOP:k=k+1:GO SUB 300: IF k<1000:EXIT LOOP
```

will not work. The exit condition is never reached. However a Procedure call will work in these circumstances. Placing a GO SUB within a short form of a FOR loop also gives funny results.

Lastly PCW benchmarks BM1 and BM6 both use empty FOR... NEXT loops. In BM1 I find:-

```
100 FOR n=1 to 1000:NEXT n
```

to be faster than the alternative

```
100 FOR n=1 to 1000 200 NEXT n
```

while in BM6 where the loop lies within a GO TO (or REPeat) loop the two line form is faster.

Finally, twice in the last couple of days, after I had finished a session playing with these ill-behaved loops, my QL crashed with a disabled keyboard. This is a phenomenon I have not experienced since the early days of using Quill v 1 and is, I hope, a result of my messing about with the Basic and not a sign of old (QL) age!

David J. S. Guthrie

I thought I would just drop you a line about the use of extra memory with Quill, now that I have a 128K DRAM card installed in my QL. You will recall Neil Crighton's letter on the subject in June '85 QUANTA (p10). Neil's suggestion that one probably did not need vast amounts of extra memory seemed very sensible, and 128K of extra memory sounded about right.

Unfortunately, it has taken about six weeks to get a working DRAM card out of PCML. The first two cards had to be sent back, because after a few minutes of use they refused to work (one would hang up and the other gave a blank screen after self-test). The card I have received this morning does work, (three hours of use so far). In an accompanying letter, Lisa Fuller of PCML wrote that the problem was with the 64K chips and the board I now have uses 256K chips.

Is Quill faster with the extra memory? The answer is a bit of a lemon, I am sorry to say. On the credit side, a large document can now sit totally in memory, this speeds up all editing operations and merging. On the debit side, <SAVING> is no faster, and perhaps slightly slower, still taking approximately 1 minute per thousand words. <BACKING-UP> within the document is actually significantly slower, this took about 1 minute per 3000 words without extra memory and now takes 3 minutes with. One can see by watching the Microdrives that the document is probably being transferred in smaller chunks.

So is the extra memory with Quill worth the £135 with VAT? My initial impression is rather unfavourable as I was particularly hoping to speed up Microdrive operations, but I will be writing again with a more considered opinion in a few weeks.

Dr John Heckmatt,  
17, Cleveland Court,  
Kent Avenue,  
Ealing,  
LONDON W13 8BJ.

#### **PROBLEM SECTION**

Has anyone discovered the secret of how to force Quill to print a footer on the last page of a document? I find that this works sometimes, but not always - for example, one document will happily print "Page 2" at the foot of the final page before it finishes printing, whereas another will stop output to the printer when it reaches the last line of text, leaving (as it happens) Page 7 without a footer. I can't see any reason why this should be.

Is anyone else frustrated by the lack of a case-change facility in Quill? This used to be very simple in Wordwise on the BBC micro - just position the cursor at the start of the appropriate text and key <CTRL> <5>. The cursor would run through the text, changing upper to lower case and vice versa, until the keys were released. Quite

apart from the speed of cursor travel, which is much faster in most applications in Wordwise than it is in Quill, this was a useful feature which I often miss. Is there any way of achieving this which I have yet to discover? If not, would anyone support me in asking Psion to incorporate this feature in an upgraded version of Quill?

My QL (D09, QDOS 1.03, JM SuperBASIC, Psion 2.00) is fitted with the PCML 256K RAM expansion. When working in Abacus, the display shows memory availability at first to be around 233K, but as the spreadsheet fills up the available memory suddenly drops to around 62K. I haven't yet managed to notice exactly when this occurs (I'm usually concentrating on another part of the screen at the time). Can anyone explain why this happens (in simple terms, please - I'm no expert)?

When switching on my system, I usually adopt the sequence of mains, printer, monitor, QL, with the reverse sequence when switching off. All hardware is plugged into an RME 4-way spike-suppressed extension lead which is in turn plugged into a switched 13-amp wall socket. Having to control power to the QL by plugging in or unplugging the lead from the power supply to the computer seems to me to be not only undignified but also likely to wear out the plug and/or socket over a period of time. Has anyone any suggestions for a method of operation, or a hard-wired switch in the power supply lead, which will allow me literally to switch the QL on and off without damage to it or the peripherals?

Paul Gelling,  
Bank House,  
Ripple,  
Tewkesbury,  
Glos. GL20 6EP

/\* The problem with a footer on the last page of a document might be due to the printer not printing the buffer contents when Quill finishes. I think you might be able to get round this by printing a final CR, which should clear the buffer. You can probably do this by changing the postamble codes.

My system is supplied through a 4-way mains adaptor, and I always switch everything on or off together at the wall socket, without any problems. \*/



### **MAKING SPACE ON THE ARITHMETIC STACK**

My problem with this was published in issue 4 (May 1985), but would like to point out that I differ with the explanation given in the footnote. i.e. the /\* \*/ that sometimes follows an article. My problem was with the BV.CHRIX routine which one has to invoke in order to obtain space on the Arithmetic Stack. As I was getting non-zero error returns in D0, I felt the only way to prove that anything was happening at all was to request inordinately large amounts of space. There was no "response" even for 64K bytes, but everything froze when 128K was requested.

I have since had the opportunity to compare Adrian Dickens' Guide with the QL Technical Guide and to take the best of both worlds. The Technical Guide specifies that D0.L should hold the number of bytes required, while Adrian Dickens specifies DI. So I have included a MOVE.L D0,D1 within my code, but on return D0 never seems to be zero. Also, contrary to the advice given, I have managed to set up on stack a string of up to 12K bytes and return this to Basic as a function result without any ill effects.

Bill Cowhig has asked a somewhat related question in his article which appears above mine, and is concerned with how the Arithmetic Stack could be tidied up. I would like to suggest to him that when he comes through from BASIC he should immediately store the value of BV.RIP(A6) somewhere safe within, say, his RESPR area, and then all he has to do to tidy up is to refresh BV.RIP(A6) with this value before returning to BASIC. As I understand it this value is a relative one and so is valid wherever the Arithmetic Stack happens to end up.

Ivan Krishnan,  
12, Giles House,  
160, Westbourne Grove,  
London W11 2RJ.

### **MICRODEAL GAMES REVIEW**

#### **Hopper (£14.95)**

The first is a Frogger clone, what can I say that hasn't already been said about Frogger.

The frog looks like a frog, the cars look like cars. It is a fast 100% machine code game (like all of Microdeal's other software). The game has three different levels; first is crossing the busy road to the river bank then, across the deteriorating logs to home. After getting all five frogs through this, comes the second stage, same again but with a snake on the river bank to hinder your way, third is (yet again) the same, but with a crocodile thrown in for good measure.

If you liked Frogger on the Spectrum you'll love Hopper on the QL.

### **Cuthbert In Space (£14.95)**

Yet another winner from Microdeal's Steve Bak. This is the kind of game (that for my money) made shooting aliens popular in the first place. After saying that the graphics hardly do the QL justice, being more Spectrum than QL standard. The game itself has nine levels (each with a different screen), and then go round again starting at screen one with more baddies to shoot.

This is one of those games that have that certain feel, the one that makes you sit up until two in the morning zapping. Everyone I know now has the Cuthbert habit.

### **Land Of Havoc by Microdeal (£19.95)**

It's the one I've saved for last that to my mind is the best of all the QL games I've seen to the present time. Lands Of Havoc is a convert from the Commodore 64 It won the game of the month award from two well known Commodore wags. and the accolade was well deserved.

The plot is that you are 'Sadar' the lizard man , saving the Land Of Havoc from the invading Dark Lords and their evil minions. The game has three totally different levels; first is the Land Of Havoc where you have to find certain icons (while defending yourself against the forces of evil) in order to get to the second level, Land of the Trolls, after collecting the seven talismen you can then enter the portcullis into the land of the Dark Lords. Kill them and your task is over. The game has over two thousand screens, hundreds of Daddies and a well thought out plot.

At £19.95 it's not a cheap piece of software, but it is excellent value for money in terms of amusement, continued enjoyment and is a must for any budding graphic adventurer.

All three of these new games are professional/quality software, It's nice to see some inventive games software coming out for the QL at long last. I for one have placed a standing order for all new material from Microdeal.

David Storton

### **THE NAME TABLE (ENTRY FOR A PARAMETER)**

Both Adrian Dickens and the QL Technical Guide agree that extra information is masked into the Name Table entry (Bytes 0-1) when it represents a call parameter. However they seem to differ on which byte it is of the two. The Technical Guide shows everything back to front in its table of entry types thereby making things more confusing than ever.

To resolve this I took a hex dump of the Name Table entries for two call parameters before and after a CA.GTxxx routine:-

Before 02120073 00000010 02010083 00000038  
After 02120073 00000010 01010083 00000038

A3 & A5 were set to point to the second entry and CA.GTSTR was called. As can be seen there is a change to the entry, namely in Byte 0 which changes from '02' to '01'. Both Adrian Dickens and the Technical Guide state that these routines smash the bits which indicate what the separator was in the original call. The former states that these bits are in Byte 1 and the latter that they are in the upper four bits of Byte 0, but it turns out to be the lower four bits of Byte 0 which are affected. So is it that both sources are wrong or is it that the vector routine is smashing the wrong bit?

Ivan Krishnan,  
12, Giles House,  
160, Westbourne Grove,  
London W11 2RJ.

### TRANSLATIONS AND THE JUKI 6100

A response to some pieces about using printers in July's QUANTA.

Harold Bennett was having trouble printing both £ and ` signs. It's really quite simple, and it sounds to me that he was trying a more complicated way than is necessary. If your wheel has a £ sign on it, then it is an American wheel. So set the Juki for the American character set, viz dip switches 4,5,6 all off (don't forget to change dip switches with the power off to be effective).

Press the £ key, and the printer will print it. To get the ` use TRANSLATE -, ESC, I. That's all there is to it.

Well, you can go on.... The QL code for '£' is 96, but the Juki produces this with code ESC,I . So what does it produce for code 96? With my daisywheels, it's either a grave sign or tiny circle. But how to produce code 96 on the printer? Translation is necessary, but from what? David Nowotnik suggests sacrificing a little used symbol like {, but this isn't really necessary. There are all the symbols on the QL from code 127 to 191 which can be used. For how to key these in see 'character set and keys' of 'Concepts'. You can get a grave sign on the screen, for example, by pressing together CONTROL, SHIFT and the / key. (This character has code 159). So, using TRANSLATE ,96, a grave sign on the screen gets printed out on the paper.

Besides the £ sign, the Juki has 5 more characters reached by escape codes. The method above can be used to get at them. For the paragraph sign, for example, translate QL code 182 (keyed in by CONTROL,SHIFT,V) into Juki code ESC,H. A last observation on translations for the Juki. Ernest Palfrey gave some suggestions in the last issue. But they may mislead some people. He suggests a preamble code of ESC, RS, ENQ. As I read the manual, this will give a very narrow line spacing.

Most people will prefer to use the dip switch to adjust line spacing. No preamble code is particularly necessary for the Juki. If we're writing stuff for beginners, let's keep it simple. (And baud 2400, parity mark may be appropriate for his system, but if you're using the popular Miracle Systems interface, baud 9600 with no parity works).

Robert Matthews,  
19, Trent View,  
Marton,  
Gainsborough, DN21 5AG.

### A BASIC EXTENSION FUNCTION BUG

If one or more extension functions appear within a summation of terms in either a print statement or on the right hand side of an assignment statement then when the print/assignment is made it is only the value of the last extension function and any terms following it that comes across, all other preceding terms having been ignored. For example, using assignments with the function XI((hex string>) which converts hexadecimal values:-

```
u=20: a=u+xi('lf'):print a -> 31
u=-20: a=xi('lf')+u: print a -> 51
```

and also:-

```
print xi('3f')+20+xi('lf') -> 31
print xi('lf') +20+30 -> 81
print xi('lf')+20+30+xi('3f') +40+50 -> 153
```

Using parentheses the results are even weirder as also are those with other operators.

Ivan Krishnan,  
12, Giles House,  
160, Westbourne Grove,  
London W11 2RJ.

### LIBRARY CORNER.

There have been several updates and additions to the library recently and we now have 21 cartridges of programmes and utilities and expect to add another (Library 16) by early September. DASM\_XDR has been updated and extended and contains a boot programme which will select the device to be used. A programme well worth adding to other cassettes. There are now 2 cartridges to the Library guide, the first covering library cartridges 1-12 and the second from 'Demo' onwards. Both guides contain details of alternative methods of ordering library programmes and a list of sub-librarians.

Recent additions include QDRAW (£1 option), FILE\_DOCTOR and QUILL\_DOCTOR from Tony Towers, an excellent screen compression programme from Ron Dwight (£1 option), and a very interesting Turtle drawing programme from David Coles. In the pipeline is ARTIST from Philip Murphy and SCRIPHID (CAD) from Ted Cogswell plus a number of other very useful programmes.

Your sub-librarian has details of all the programmes in the library on the two libguides which are obtainable from him by sending £1.50 for each, plus a stamped addressed Jiffy bag, a formatted cartridge and label from the envelope that QUANTA came in). Cheques should be made out to IQLUG. If you do not know where to contact your nearest sub-librarian telephone (0708) 27272 for details.

If you do not wish to supply your own cartridge they can be supplied direct from me by adding £2.00 per cassette. This will include the cost of postage and packing but you should still include a label. You may also order by Transcash at any Post Office to the Library Giro account 35 032 4905 or by Giro transfer to the same account. The library depends on contributions from members and we have been very fortunate in both the quantity and quality of the programmes submitted. If you have a programme that you have written that could be of use to other members please send it to Brian Davies, 72, Wharnccliffe Street, Hull, Humberside, HU5 3LY for submission to the library or constructive criticism.

S. Day,  
Librarian,  
241, Highfield Road, Romford, RM5 3AW.  
Telephone (0708) 27272

#### **68000 ASSEMBLER?**

Ivan Krishnan, an experienced assembler programmer who has worked for computer manufacturers, is interested in taking on 68000 programming assignments from software houses, etc. If required he will also provide separate accompanying documentation to whatever level of detail deemed necessary.

Ivan Krishnan,  
12, Giles House,  
160, Westbourne Grove,  
London W11 2RJ.

#### **STATISTICS**

I have written two programmes in BASIC, for statistical analysis. The first is for quantitative comparisons. i.e. you want to make comparisons between two measurable parameters (such as height or weight of children from two different schools).

The second one allows comparisons between qualitative and quantitative parameters. For example, the height of boys with blue eyes in comparison to those with brown eyes. Of course you have to choose what kind of comparisons you want to make.

Both programmes run in the same way. You first have to create files of values (of course 'quantitative' and 'qualitative' files are not compatible) and save them on a cartridge (drive 2). Every file can be subsequently modified (i.e. you can add values without retyping all the previous values).

The programme devoted to quantitative statistics offers you two possibilities:

1. analysing a series, and
2. comparing one series to another.

The first option evaluates the series. Values are ordered (and you are invited to save the ordered series to save time for further readings) and classified by groups with the same range of extreme values. Parameters are calculated: arithmetic, geometric (when possible), harmonic (when possible) means, median and mode. Then you can obtain the standard-deviation, various 'moments', and several coefficients that evaluate the Gaussian distribution (or not) of this series.

The second option allows you to compare two series. You can use parametric tests, if your two series can be considered as normal, or non-parametric ones. Also, you are asked if your series are paired or not. I have included a control test which disallows the calculation if you say that the series are paired when they do not have the same number of values.

You have the choice between correlated and uncorrelated Student's t test, Mann-Whitney test and Wilcoxon test. As you may know, non-parametric tests 'don't like' ties and it is better to discard such values. If you don't, you'll surely decrease the strength of the tests and maybe obtain an error message (due to division by 0), but you can try...

The 'qualitative' test allows you to compare your series and choose the best test for this purpose (chi-square tests or exact Fisher's test). The results are displayed on the screen, and you only have to make a note of them.

Bernard Dalens,  
9, Chemin de Villard,  
63670 Gergovie,  
FRANCE.

### **STARTREK**

This is a game which I have written in BASIC (and in French) for my children. They enjoyed it. Maybe some of you also like it... You are Captain Kirk, at the head of the USS Enterprise. Rations and Klingons are attacking your sector of the Galaxy.

They have put lots of hostages into jails on several planets. In some others, they have stocks of energy, fuel and jewels.

Your mission is to destroy these enemies, save the hostages and discover the stocks for your own use. Your ship has energy which you can (and must) use for your moves, your shields (don't forget these or you'll be destroyed at the first attack), and for your laser. You also can use missiles to destroy Klingons and Romulans, but you must be clever with your hands ...and you have only 20 missiles to shoot.

There are some star bases in the Galaxy. You can obtain considerable help from these: crew, energy, fuel, repair of the engine, laser etc., provided that your landing is correct.

The Galaxy comprises 64 sectors but you only can have (command 'SCANNER') a sight on 9, with the Enterprise right in the centre. You must never leave the galaxy or you'll be destroyed. Star bases are shown in yellow, whereas enemies are in two kinds of red.

To have contacts with enemies, star bases, planets... you must use the command 'RADAR' and obtain a sight of the present sector. You can find enemies shooting at you (lasers only, which progressively destroy your shields) and coming close to you. You can also find natural (black holes, desperately attracting your ship) or artificial (magnetic mines coming close to you) obstacles.

To escape, you must, very quickly, press the right command, i.e. dematerialisation or hyperspace ...

When landing on a planet (be sure not to crash), several events can occur, depending on the area where you land. An appropriate decision is the sole way to benefit by this. Remember that several planets are partially or totally opposed to you (for natural or other reasons).

Have a good play and improve the game....(I would be very happy to see some of these improved versions).

Bernard Dalens,  
9, Chemin de Villard,  
63670 Gergovie,  
FRANCE

## **SOME ARCHIVE PROGRAMMING APPROACHES**

### **ARRAYS**

There have been times when the need for arrays has arisen, and one has had to consider the alternatives:-

1. Set up a record to hold the various values, and use the FIELDV() function to extract values. But then dynamic updating of the values becomes a bit cumbersome.
2. Implement an array facility using Archive procedures.

As it was also required to perform updates of the array elements the second approach was adopted. This is best explained with an example.

Supposing it is required to set up an array t with n elements whose values fall within the range 0.00 to 99.99, this could be done by setting up each value as a 5 byte string, with leading zeros where necessary, and then concatenating.

The procedures that follow are given with the corresponding SuperBASIC statements in the right-hand column.

Archive Procedures	SuperBASIC Statements
	Initialise
let t\$=rept('0',5*n)	dim t(n)
	Insert a Value
let z\$=str(z,0,2) let pf\$=' ': let sf\$=' ' if len(z\$)<5:let z\$='0'+z\$:endif let y=5*i: let x=y-4 if x>1: let pf\$=t\$(to x-1):endif if y<len(t\$): let sf\$=t\$(y+1 to endif: let t\$=pf\$+z\$+sf\$	t(i)=z
	Extract a Value
let y=5*i: let x=y-4 let z\$=t\$(x to y): let z=val(t\$)	z=t(i)

### TAGS & LOOPING

There is no 'goto' in Archive, but one might sometimes wish there could be. The way round this is to set up one's own tags or labels. This is however closely intertwined with the fact that one cannot easily 'exit' out of a whole series of nested 'if' and 'while' loops. The method used for these is to set them apart in a separate procedure and to use the 'return' statement. This will be assumed in the explanation of the following flow-chart.



```
Begin
::
Open file
::
Select record
::
Inspect and amend it
::
Error during amendment?
N ::
Update the record
::
More records to select?
N ::
Close file
::
End
```

The above application could describe a situation where there is a procedure 'sel' which processes a file serially looking for a certain type of record. When a record is selected the inspection routine 'inspct' is called. The inspection routine allows amendments to be made if required. After this there is an opportunity for further amendment if the record has been incorrectly amended. The record in the file is then updated and the process is repeated until the whole file has been processed

The tags that will be used for these loops will be set up as values of the string tg\$. The string 'sct' will be used to loop to the selection procedure 'sel', and likewise 'Ins' will be used for the procedure 'inspct':-

```
let tg$='sct': while tg$='sct': sel: let tg$='Ins'
while tg$='Ins': inspct: if corr:let tg$='Ins': else
let tg$='sct': endif: endwhile
if not more: let tg$=' ': endif: endwhile
```

It is assumed that the procedure 'inspct' also sets up the global variable 'corr', which then acts as a re-entry switch to 'inspct'. Also the global variable 'more' is set by the procedure 'sel' if there are more records left to process.

Forward jumps can be accomplished by enclosing non-relevant code within 'if' or 'while' blocks. A null value for tg\$ would assume linear flow. For instance, a forward jump to a block of code labelled 'lab1' could be engineered thus:-

```
if tg$=' ': let tg$='blk1':. : endif
if/while tg$='blk1':. . . : endif/endwhile
if tg$=' ': let tg$='blk2': endif
if/while tg$='blk2':. . . : endif/endwhile
. . .
if tg$=' ': let tg$='lab1': endif
if/while tg$='lab1':. . . : endif/endwhile
```

The last elaboration there remains to suggest is where several routines are concerned and it is required for 'control' to 'float up' through several routines to the right tag in the appropriate procedure. A device which allowed this would assume each procedure to have a unique, local identifier string, say id\$, which could contain a string like 'p<number>' or similar. Therefore, on return from a called procedure, only tags beginning with the identifier string have currency within this procedure, otherwise control 'floats up' to the next level of call:-

```
local id$='proc5'  
...  
if id$=tg$: let tg$=' ': endif: let li=len(id$)  
if tg$<>' ': if tg$(to li)=id$: let tg$=tg$(li+1 to): endif  
else: return: endif
```

On return from the called procedure, tg\$ is stripped of the id\$ string, otherwise exit is made to the higher level procedure, etc.

#### **STRING SPLITTING ROUTINE**

It was quite pleasing to discover how a quite complicated flowchart reduced to just a few lines of Archive.

The problem which one is always meeting up with is that of splitting a string correctly when it extends over more than one line.

The solution is to split the string on either a blank or on a character immediately preceding a blank character. If the end of line does not coincide with either of these two conditions, a backward scan is made until a blank character is found, whereupon the string is split and the next character, being non-blank, begins the next line.

If there was no need for a backward scan the remainder of the string will only be printed if it contains non-blank characters. Also, the next line will begin with the first non-blank of the remaining characters. In other words leading blanks are ignored.

The procedure that follows assumes a physical line limit of 64 characters, and prints the string p\$, splitting it into lines which are of length 11.

```
proc split; p$,11  
local z,s,l,lp  
let lp=len(p$): let s=1  
while s<=lp: let z=s  
if z+11>lp: let l=lp-z+1  
print p$(z to z+l-1)+rept(' ',64-l);: return  
endif  
let l=11
```

```
if p$(z+l)<>' ': while p$(z+l-1)<>' ': let l=l-1:
    endwhile
    let s=z+l
else: let s=z+l+1
while s<=lp and p$(s)=' ': let s=s+1: endwhile
endif
print p$(z to z+l-1)+rept(' ',64-l);
endwhile
Endproc
```

### LOGICAL OPERATORS

In Archive parentheses seem to have no effect in determining how a logical expression should be evaluated, in fact their use leads to an error being flagged. Logical expressions appear to be simply evaluated from left to right. As an example, if one was trying to check for opening and closing parentheses within a string, it would be convenient to make the following statement:

```
if (u$='(' and v$=')') or (u$='[' and v$=']') or (u$='{ ' and v$='}')
```

The statement in the example above is not acceptable, and so a series of switches has to be set up to hold the parenthesised items, viz:

```
local sw1,sw2,sw3
let sw1= u$='(' and v$=')':
let sw2= u$='[' and v$=']'
let sw3= u$='{ ' and v$='}'
if sw1 or sw2 or sw3: . . . : Endif
```

Ivan Krishnan,  
12, Giles House,  
160, Westbourne Grove,  
London W11 2RJ.

### USING QUILL

Leon Heller suggests (QUANTA vol.1 issue 11 p.33) that forced page breaks in Quill version 1.03 can be removed by taking the cursor to the line below the page break, then (using the up arrow) to the page break itself, then deleting the line with <CTRL> <right arrow>. To make this work in Quill 2.00, use <CTRL> <left arrow> instead.

Paul Gelling,  
Bank House,  
Ripple,  
Tewkesbury,  
Glos. GL20 6EP

### LOADING D-DAY

Regarding the problem about loading D-Day. If you place a piece of sticky tape over the place where the write-protect lug used to be on mdv2, then it should load correctly. (It takes a long time!!)

This information was obtained by phoning Games Workshop, who were very cooperative.

Paul Kutchera,  
Little Trottenden Oast,  
Lidwells Lane,  
Goudhurst,  
Cranbrook,  
Kent TN17 1EJ.

### **"AT" KEYWORD CO-ORDINATES**

Francis Cameron (QUANTA vol.1 issue 6 p.27) noticed that the "AT" keyword worked differently after his machine was replaced; I think the answer lies in the SuperBASIC version, and the clue to this can be found in a procedure which appears in the Psion install\_bas programme (and others in the Psion suite). The procedure goes like this:

```
10 DEFine PROCedure PT(a,b)
20 IF ver$="FB" THEN
30 AT a,b : ELSE : AT b,a : END IF
40 END DEFine
```

and is called elsewhere in the programme with statements in the form of (e.g.) "PT col,line".

From this I deduce that the FB version uses "AT" with the column number as the first parameter, and the line number as the second, whereas (presumably all) the other versions use "AT" with line first, column second (as described in the manual). It seems a shame that one version should be so radically different that a routine like the above has to be incorporated in software in order to make the BASIC work properly. Can there be any explanation other than sheer carelessness?

Paul Gelling,  
Bank House,  
Ripple,  
Tewkesbury,  
Glos. GL20 6EP

### **INSTALLING THE EPSON MX-80 PRINTER**

Rather than boring the vast majority of QUANTA readers with a blow-by-blow account of how I installed my old Epson MX-80 III, using the Miracle Systems serial-to-parallel converter, I will just say that I have achieved the following with no real difficulty and I would be happy to advise anyone who needs help.

Easel screen dumps work perfectly without "stripes" caused by superfluous line feeds.

Programme listings appear with the correct linefeeds and with a hash in the hardcopy where a hash appears on screen.

Quill, Abacus and Archive produce the required results when files are dumped to the printer.

I can access the alternative styles supported by the Epson, such as condensed and expanded type.

I can use characters such as accented letters which are supported by the Epson in one or other of its national character sets and can also be shown on the screen using the QL character set. In other words, if an e "acute" appears on the screen, it will appear quite correctly in the hardcopy. This also applies to the pound sign.

I haven't used the characters themselves in the course of this item in case this causes problems in printing it on the QUANTA daisywheel!

Paul Gelling,  
Bank House,  
Ripple,  
Tewkesbury,  
Glos. GL20 6EP

#### **MICRODRIVE CARTRIDGE PRICES**

The cheapest Microdrive cartridges I have found are at larger branches of Boots, where packs of four are currently (end-July, early August) on sale for £7.50 and (according to one assistant I spoke to) are selling like hot cakes. The packs come complete with a wallet (like the Psion package), stickers and index slips.

Paul Gelling,  
Bank House,  
Ripple,  
Tewkesbury,  
Glos. GL20 6EP

#### **EASEL 2.00 TIPS**

Two tips which might help those who (like me) have little experience of working with Easel...

Merging two or more sets of figures from files saved on Microdrive will not work using the Load command (which performs a "new" and erases existing data); there is no Merge command as such. The answer is to use a roundabout method of export and import, as follows:- Suppose you have two Easel files on Microdrive called "sales" and "costs". To merge them, load "sales" then save it as an export file (using the Files command). Now load "costs" in the normal way, then (again using the Files command), import your "sales" file. Both sets of data will now be available in RAM, and can be seen together using the View command and saved together as a new file in the normal way.

Altering the appearance of text is not very clearly explained in the User Guide, but the Help file is easier to follow and solved a problem for me. The trick is to use the Change command (choose the Text option) to define a text style - ink and paper colours, etc. - and then use the Edit command (again choosing the Text option). Even if you make no changes to the wording of the text, you will find on exit from the Edit command that the appearance of the text has been changed to your recently defined style. By this method you can alter the appearance (individually or altogether) of the vertical and horizontal axis names, the cell Labels, the title, and any 'extra' text you have inserted. The key will always appear in the 'current' text style, so if you want this to be different from other text you must make your key style the last change; otherwise it will be affected by any subsequent changes to other text styles. To sum up, all text except the key) will appear in the text style which, AT THE TIME EACH TEXT ITEM WAS LAST EDITED, was the "current" style

Paul Gelling,  
Bank House,  
Ripple,  
Tewkesbury,  
Glos. GL20 6EP

#### EXTENDED PRECISION ARITHMETIC

E.G. Whitbread's letter in the July issue was interesting to me as I wrote similar floating-point routines for the late ZX 80, which was, of course, integer-only, and subsequently for the ZX81, using String Input and arrays. On the ZX 80 string-slicing was accomplished by means of CODE and TL\$ (truncate left string). Oh, happy days:

Now while the QL's arithmetical precision in SuperBASIC leaves much to be desired, and falls short of that promised in early publicity, the precision offered by ARCHIVE and ABACUS in numerical calculations is to 12 and 14 places of decimals respectively.

By this, I mean that the 12th and 14th digits (behind a decimal point) are rounded-up. Here is an example Imported from ABACUS.

<u>Column A</u>	<u>Column B</u>	<u>Column C</u>
0.5     i.e.	0.5 radians	Using B2 and B3
sin(0.5)=	0.4794255386042	instead of sin(A1)
cos(0.5)=	0.87758256189061	and cos(A1)
tan(0.5)=	0.54630248984363	
sin(0.5)/cos(0.5)=	0.54630248984363	0.54630248984363
sin(0.5)^2+		
cos(0.5)^2=	0.99999999999964	0.99999999999964
sin(0.5)*sin(0.5)+		
cos(0.5)*cos(0.5)=	1	1

ARCHIVE rounded up the value of cos(0.5) to 0.877582561891 and gave the correct value of "1" for sin(0.5)^2 + cos(0.5)^2, as one would expect from the figures above.

Using ABACUS and ARCHIVE for calculations is quite straightforward and I would not have thought special routines worth while unless of course, E. G. Whitbread needs even greater precision than this.

Ron Humphries,  
16, Coniston Road,  
Sutton Coldfield. B74 3LE.

#### **D-DAY BY GAMES WORKSHOP**

With reference to the query in the July, 1985 edition of QUANTA, we too experienced a similar problem. I purchased a copy of D-Day from Logic Sales Ltd. at the ZX Microfair in June. Upon returning home, my son expectantly tried to load the game only to gain the same result as you experienced - namely 'bad or changed medium' or 'end of file' error messages. On the Monday morning I phoned Logic Sales, explained the problem and they suggested that I might have a faulty programme. They readily agreed to exchange the cartridges which they did very promptly by return of post. Once again we placed the cartridges in the Microdrives ready to win World War II all over again - but once again up came 'bad or changed medium' or "end of file"!! The nearest we came to doing battle was the language, which was more akin to the barrack-room!!

Next day I rang Logic Sales again with the same sad story, only to be told that they had tested the software on their QL before they sent it to me and that it had loaded properly; they had also sold a number of copies at the ZX Microfair and I was the only person to have indicated such a problem. They suggested that it could be the dreaded Microdrive incompatibility problem and that perhaps I could try the programmes out on a friend's QL. As none of our friends have a QL, I explained that this was not really possible, and anyway, we have not experienced problems loading any other software. However, they did suggest that my best bet was to contact Games Workshop to see if they could supply a solution. This proved to be the most constructive piece of advice to be given yet. (Having got you to this cliff-hanger, I am tempted to do a 'Dallas' and leave you in suspense for another month or so! But I won't!)

I immediately rang Games Workshop and was quickly put through to the "D-Day on the QL" expert. I explained the problem, expecting to be given similar advice. But no! I was told that this problem has arisen through the company who duplicated the cartridges. To enable the game to run properly, Microdrive 1 needs to be able to write to Microdrive 2. Therefore, the Microdrive cartridge labelled 'D-Day Two' requires the write-protection lug to remain intact. It appears that a batch of these cartridges had the lug removed. So the answer was to place a little piece of sellotape over the offending gap and the problem should be resolved. I thanked the chap for his assistance and, as I was at the office, I then had to patiently wait until the end of the day before I could get home to implement the final solution!

So, armed with sellotape and scissors the gap was bridged, the cartridges inserted, F2 pressed and "GERONIMO!" - Battle commenced! Winning the War must prove easier than this!!

I hope that our experience assists any other frustrated war gamers that may be having problems loading this game. It is an expensive game and has proved to be quite enjoyable, so it would be a shame if it gained the reputation of being unworkable. As I expect most of your readers know, Games Workshop are no longer producing software for the QL and only a relatively small number of this game was issued. Therefore, D-Day could well be a collectors item.

With regard to a review of the game, I think that my young son is better qualified for this and so I have asked him to produce one, which follows below.

I would finally like to say that I joined QUANTA at the ZX Microfair and have found the magazines very interesting, although being a virtual computer novice, some of the articles have gone over my head! However, I am sure that they will come in useful in the fullness of time! I believe the QL to be a good machine and have been heartened by the recent increase in software and ancillary products. Perhaps it will now go from strength to strength and receive the recognition that it deserves. Anyway, that's enough from me, so over to my son.

#### **REVIEW OF D-DAY FOR THE QL BY GAMES WORKSHOP**

Once we had managed to load our copy of QL D-Day (this was done with a lot of trouble to say the least!!) we found that it was an enjoyable game, on the whole.

Whilst loading, which incidentally is a very long time, there are hardly any loading graphics at all. I found this made the wait even more tedious, but the only loading graphics there were (a tank) I felt were fairly good.

During loading the computer stops occasionally to ask you which of the options you would like. e.g. you can play 1 or 2 player games, the computer can be made to play as the Germans or the Allies, you can have 15 or 50 unit armies each, and you can choose between 'The Landing', 'Breaking Out', 'To Arnhem' or 'Arnhem' as your scenario.

Each scenario has its own objective (to win you have to make the objective or kill 75 per cent of your opponent's force).



**THE LANDING:-** For this the Allies have to occupy the bridge at the bottom of the map, and the Germans have to capture the town at the top.

**BREAKING OUT:-** Here you have to capture two of the three main bridges.

**TO ARNHEM:-** You have to occupy the main bridge

**ARNHEM:-** Here your aim is to capture the buildings next to the central river and between the bridges.

Whenever we have played we have not been able to quite work out what 'occupy' means, for when we win we normally win by being nearest to gaining the objective at the end of the maximum 15 game hours although we have had troops on the bridge for ages.

During the game you have to move your units around a map and fire them. The firing process can become terribly frustrating when you find that your line of fire is obstructed or that your unit is out of range. When you have finished hiding away your forces so that you are sure that your opponent cannot get at your units you end your turn, and then watch the computer rip your army to shreds! Although this game is almost a must for all war game fanatics I do think that it could do with a few improvements in some areas, such as sound.

I think that better sound effects and better movement (as this took a long time) would have made for a more enjoyable game. Generally, it was an excellent game all round.

Brian and Gareth Turner,  
41, Meadows Road,  
Cheam,  
Surrey.

#### **SENDING PRINTER CONTROL CODES IN QUILL etc.**

**N.B. WHEREVER AN @ APPEARS THIS SHOULD BE READ AS A COPYRIGHT SYMBOL. (In reformatting for the QUANTA archive,(2015) this has been corrected as far as possible - JG**

I think it would be useful to pass on the following advice particularly for people who have version 1.00 of Quill and therefore cannot translate more than 2 characters via the printer driver. Follow J R Catt's advice of December 1984 and redefine the © (copyright symbol) to ESCAPE thus:-

TRANSLATE 2: 127,27

It is then possible by typing any of the following 2 or maximum 3 character combinations (i.e. copyright symbol + 1 or 2 other characters NOT separated by any spaces) anywhere in your text to achieve any of the following effects:-

©-1 underline on  
©-0 underline off  
© 4 italics on  
© 5 italics off  
© E Emphasized on  
© F Finish emphasized  
© G double strike mode on  
© H double strike mode off  
© M set elite  
© P set pica  
© W1 WIDE (Enlarged) on  
© W0 WIDE (Enlarged) off  
© S0 superscript on  
© S1 subscript on  
© T Turn sub/superscript off  
© p1 proportional spacing on  
© p0 proportional spacing off  
© !x set print mode ( where x= 2,4,8, etc)  
etc., etc .....

Using this approach you can keep the other translation codes for those effects that cannot be achieved using ESCAPE + some character i.e. condensed and translation of international characters.

K. S. Singh,  
39, Broad Chalke Down,  
Winchester SO22 4LR.  
Tel. (0962) 65184

### **BASIC PROGRAMMING HINTS**

QL SuperBASIC supports strings of 32768 characters whilst most other computers only allow strings of 255 characters. SuperBASIC also has very powerful string slicing techniques. A combination of those two factors gives SuperBASIC a simple but effective method of carrying an ordered index to an array or arrays. Indeed such an index to an array need never be out of order, so that a fast search can always be accomplished. Let me demonstrate the technique. I will keep it simple using a one dimension alphanumeric (or string) array which however has internal fields. Assume you are keeping your accounts on your computer and you have the following:-

Date (6 numerical characters, 2 each for day, month and year) i.e. [ddmmyy]  
Ledger (4 alphanumeric characters) i.e. [aann]  
Details (20 alphanumeric characters) i.e. [aaaaaaaaaaaaaaaaaaaaa]  
Amount (10 numerical or sign characters) i.e. [nnnnnnnnnn]

The limit is only restricted by your imagination and requirements, but I will stop there.

Assume you make a maximum of, say 80 entries per month. You will need a string array of 80 x 12 or 960 elements each 40 characters long and so will have to dimension your array as, say, DIM accounts(860,40).

Each element of the array will have the details, in the order, given above so that, for instance, you know that the ledger is given by the 7th to 10th inclusive characters of each array. Suppose you want to keep an ordered index chronologically on the date; by ledger; alphabetically on the details; and in ascending order of value. You can, of course, have whatever order you like, on one, two or more fields, although the coding gets more difficult. I propose to illustrate the concept only on the date field.

You will need to know the earliest and the latest date with which you will be dealing. The first can easily be 000000 whilst the last can be 999999 (although at the turn of the century additional problems will arise). Your entry, which is in the form ddmmy in the account\$ array, will need to be put into the reverse form yymmdd so that it can be ordered. Simple coding achieves that result, viz.

```
1000 DEFine PROCedure CHANGE DATE olddate$
      1010 newdate$ = olddate$(5 to 6) & olddate$(3 to 4) olddate$(1 to 2)
1020 END DEFine
```

And it is called by, for instance, CHANGE\_DATE temp\_account\$(1 to 6)

You will first need to determine the size require for the index. For a date you will need, say, 6 numeric characters for the reverse value yymmdd plus an alpha character for the occasions when there will be more than one entry on each day. That gives you a total of 7 characters and a theoretical 26 entries on any one day. There are 960 elements in the array and thus three numeric characters will be necessary to determine to which array the index is pointing. The index will thus need 10 characters for each date entry and will be in the form yymmddann where yy represents the year, mm the month number, dd the day, a the alpha character necessary if there are two or more entries on one day and nnn the number in the array.

The index then needs to be set up with the minimum ("000000a000") and maximum ("999999z999") data positions. Thus dateindex\$ ="000000a000999999 z999" is the initial string of the index before any entries are made. You are now ready to insert data into the array and, sensibly, you will not use the first or "zero" array or else you will run into confusion with the minimum data position. If the array number is less than 100 you will need to convert it to the form nnn, and that is easily done by the following code:-

```
1500 num$ ="000" & num : n = LEN(num$) :num$ = num$(n-2 to n):
REMark 'num' is the array number
```

Each time you make an entry, at the moment when you have validated all the separate fields and are ready to accept the entries to those fields, you insert the array number (num\$) and the reverse date (newdate\$) now concatenated and called newdatenum\$ into the date\_index\$ string at its correct ordered position.

Thus, assume you have entered your data, validated it and are ready to accept it into the 1st element of the array and you want also to insert it into the date\_index\$. This is very easy since there is only one place it can go, immediately between the minimum and the maximum data positions. Assume the date is 07.08.85 which would appear in the array as 070885xxxx etc, then by using the CHANGE DATE procedure and since it is the first entry altogether newdatenum\$ will be converted (by your programme) to 850807a001 and inserted into date\_index\$ so that is now 6000000a000850807a001999999z999°.

The code is fairly simple for the first entry so I will illustrate the principles for the 123rd entry made on the 7th August 1985. You first set your newdatenum\$ string to the form yymmddznnn where 'nnn' is the array number, in this case 123 so that it reads '850807z123' and call it the first seven characters temp\$. Notice the 'z' as the 7th character. You then divide the length of the date\_index\$ string by ten to find out how many entries there are in the index including the minimum and the maximum. You then set min\_posn to 1 and max-posn to the number of entries, i.e. min\_posn = 1 and max\_posn = LEN(date\_index\$)/10. You then find the mid position between max\_posn and min\_posn and express the result as an integer, i.e. mid\_posn = INT((min\_posn + max\_posn) / 2 ). You have now found the middle position in the string. You then use a function to find the 10 characters of each block in the date\_index\$ string. I have called that function place(n) where 'n' is the first character of each block of 10 characters.

```
4950 DEFine FuNction place$ (n)
4960 RETurn date_index$(n*10+1 TO n*10+7)
4970 END DEFine places
```

Remembering that temp\$ is allocated to the first 7 characters of newdatenum\$ you compare temp\$ with place\$(mid\_posn). If temp\$ is less than the middle position tried you change max\_posn to posn\_tried and you repeat the exercise in a loop, but based on the mid position of the lower part of the date\_index\$ string until you get a comparison where temp\$ is greater. If temp\$ is greater than the middle position tried you change min\_posn to posn\_tried and this time repeat the exercise in the upper part of the date\_index\$ string. You continue the comparisons until the difference between max-posn and min\_posn is equal to or less than 1 or the comparison is exactly equal (which will only happen if you have already 26 entries for that date). You have then found where to insert your newdatenum\$ entry and you mark the place with 'mark\_posn'.

Now the reason for having the 7th character of temp\$ 'z' appears. Unless there are already 26 entries on that day it must necessarily be the most recent entry position found for that day, since the entry below will have a letter smaller than "z" whilst the entry above will be at a later date. You then find whether the first six characters of the block of ten characters immediately below your entry position are the same as the first six characters of temp\$, in which case there are other entries of the same date. If the first six characters of the block of ten immediately below are different to the first six characters of temp\$ then you change "z" to "a". If they are the same then you find what is the 7th character of the block of ten characters immediately below and change the 'z' of newdate\$ to the letter of the alphabet immediately following such 7th character. Thus if there are already, say, three entries for that day so that the immediately preceding block of ten characters has the letter "c" as its 7th character, you change "z" to 'd'. You then insert newdatenum\$ at that place.

```
4980 :
4990 DEFine PROCedure FIND PLACE
5000 LOCal min_posn, mox_posn, mark_posn, posn_tried, loop, temp$, a$
5010 l = INT(LEN(date_index$)/10)
5020 min_pposn = 1 : mox_posn = 1 : temp$ = newdate$(1 TO 7)
5030 REPeat loop
    5040 posn_tried = INT((min_posn+mox_posn)/2)
    5050 IF temp$ < place$(posn_tried) : mox_posn = posn_tried
    5060 IF temp$ > place$(posn_tried) : min_posn = posn_tried
    5070 IF temp$ = place$(posn_tried) : mark_posn = posn_tried EXIT
    loop
    5080 IF mox_posn - min_posn = 1 : mark_posn = min_posn : EXIT loop
    5090 IF mox_posn - min_posn = 0 : mark_posn = min_posn : EXIT loop
5100 END REPeat loop
5110 IF temp$ > place$(mark_posn) : mark_posn = mark_posn + 1
5120 a$ = place$(mark_posn - 1)
5130 IF a$(1 TO 6) = newdate$(1 TO 6)
    5140 a = CODE(a$(7)) : a$ = CHR$(a + 1) : newdate$(7) = a$
    5150 ELSE
    5160 newdate$(7) = "a"
5170 END IF
5180 date_index$ = date_index$(1 TO mark_posn * 10) &
newdatenum$ & dateindex$(mark_posn * 10 + 1 TO )
5190 END DEFine FIND PLACE
```

Having created the date\_index\$ file to hold all the date items in the array in chronological order then to find the items on any particular date is very simple. Further the date\_index\$ file can either be created each time the array is loaded, or it can be saved. There is a slight problem with saving it. SuperBASIC only permits strings of less than 256 characters to be save at a time, so it will be necessary to slice the date\_index\$ file into 250 character blocks and save each block separately and load each block separately and concatenate them. Notice that using 10 characters per array the maximum size of the array that can be treated in this fashion is 32768/10 or 3,275 approx. Of course this can be increased by having additional date\_index\$ files called, perhaps, date\_index1\$ etc.

The possibilities are only limited by the size of memory of the QL or its storage medium, and your imagination.

The Hon. W. D. R. Spens,  
Marsh Mills Cottage,  
Over Stowey,  
Bridgwater,  
Somerset TA5 1HG  
Tel: (0278) 732542

### **STOLEN!**

We recently had stolen from our offices a QL, serial No. D06006163, together with an Epson MX80 printer and a Philips TP200 monitor.

We would be glad if you would publish the above details and serial number, in case anybody tries to sell it or obtain the necessary cables.

Robert V. Platt,  
Eurospin Textiles,  
Canal Bank,  
Monton, Eccles, Manchester M30 8AA. Tel: 061-789 7983

### **DISC CORRUPTIONS**

Where I work we have CST 5.25" disc drives on a QL, and we have experienced an alarming number of disc corruptions and worn-out floppy discs, so we do not believe these drives to be very good. At first we had an early disc ROM, version 1.02, which we updated to version 1.10 after first getting discs corrupted, but the later ROM made no difference in this respect. The corruptions manifest themselves when a "dir" listing is taken, with continuous garbage displayed following the list of files. Sinclair were informed of this and noted that they had found the CST drives to be less reliable than others, which is why they have "badged" the MicroPeripheral drives. As for the worn-out floppy discs, it could well be that the brand we were using was not of good enough quality, and we have changed to a superior brand.

Trevor Cook,  
8, Sheridan Terrace,  
Whitton Avenue West,  
Northolt,  
Middx. UB5 4JS

*/\* Or The drives you are referring to must be Computamate's. No other members have had problems with them - they are TEAC units which have a good reputation for reliability, although they are a bit noisy. Have you contacted Computamate or CST about this problem? \*/*

### REVIEW OF TALENT'S "CARTRIDGE DOCTOR"

Every QL owner's nightmare - "bad or changed medium" - need no longer cause you to break out in a cold sweat. At last someone has come out with a comprehensive and useful package to enable corrupt files to (almost) be recovered.

"Talent Computer Systems", rumoured to be strongly associated with Strathclyde University, have applied principles of artificial intelligence in the design of "Cartridge Doctor". The programme will quite happily recover most corrupt files entirely automatically, reassembling sectors in the correct order even if the cartridge's main directory header is corrupt.

You need not know anything about the structure of data on Microdrive cartridges. But if you do and if you want to hack around with what's on the cartridge "manually" as opposed to letting the Doctor operate automatically for you, then all the facilities are there. You can patch files and their headers in ASCII or HEX, and the contents of each data block in the file are displayed in a convenient 32 x 16 character format.

The programme works sensibly using 2 cartridges - one will normally be the corrupt cartridge, and the second will be the cartridge you are using to "save" patched data onto. Nearly all "text only" and "SuperBASIC" files can be recovered with perhaps at most a page of text needing retyping; "Machine code" files are not so easy. If a sector in the file is corrupt, there's nothing the "Doctor" or you can do to regenerate the missing bytes of code.

I suspect I received a preliminary copy of the manual - mine was A5 size, and did not fit inside the package containing the "Doctor" cartridge. The 22 pages of instructions are good, but would have benefited with a good "contents" or "index" section - I kept getting lost!

The only (very minor) defects I could find were firstly that when reading the last block of a file the error message 'READ BAD' was printed, despite the fact that the last block was read correctly.

Secondly, the file transfer operation between cartridges is fairly slow (taking about 20 mins to copy a full cartridge).

Thirdly, it would have been nice if a display option had been available to display in graphical form a map of all sectors on the source cartridge, showing in red the "BAD" and white the "GOOD" sectors together with a facility to print the sector addresses of bad sectors. This would allow manual identification and recovery of faulty data to be performed more easily, and could be performed independently of file structures on the cartridge.

Despite these very minor weaknesses, the "Doctor" represents very good value for money at £14.95 for people (like me) who cannot yet afford or justify the acquisition of disks.

I have already used the programme to recover text files from a cartridge I dropped (the tape creased - not recommended), and can recommend the programme to anyone else who has problems controlling themselves when handling Microdrive cartridges.

Rob Sherratt.

#### **AN ADDITIONAL NOTE ON MICRODRIVE RELIABILITY**

If your Microdrives are playing up regularly, then you should return the machine to Sinclair with a note of the problem. Present machines are very reliable. To quote from the QL Service Manual, "the Microdrives have been the source of some problems in the pre-issue 6 units". So if your QL circuit board is issue 5 (and most machines with serial numbers prior to D11 will have issue 5 circuitry) and you have Microdrive problems, then return your machine and ask for replacement with an issue 6 circuit board.

Rob Sherratt

#### **BACKING-UP HISOFT MONQL**

My experiences of trying to make a backup copy of Hisoft's MONQL monitor are quite different from those of Roy Barber (July issue). My master copies were not self-destroyed when I tried to copy them, but the resulting 'backup' copies are useless all the same. When one tries to load them, one gets the message 'At line 380 bad or changed medium'. I believe this is because the master tapes actually contain three files, only two of which, however, appear in the directory. (An analysis of the in-use sectors shows that about 26 sectors are not accounted for by the directory). Any copying process based on directory contents is therefore doomed to failure, which is no doubt the intended effect! However, it is difficult to see how a standard copy utility can cause the files being copied to be destroyed. If I am right about the 'hidden' file on the MONQL tape, then copying anything to the tape is likely to cause loss of files.

I certainly feel a bit vulnerable having in my possession software that is uncopyable for backup purposes, although one understands the author's reasons for doing this. Certainly Hisoft are at fault for not telling people of this restriction.

Regarding MONQL itself, however, I find it an excellent debugging tool with no real omissions except possibly the ability to do hex. Arithmetic.

Trevor Cook,  
8, Sheridan Terrace, Whitton Avenue. West,  
Northolt,  
Middx. UB5 4JS



### SAVING STRUCTURES IN C

One topic that doesn't seem to be covered in any of the books on C I've come across is how to save structures to disk (or Microdrive). The following programme illustrates the technique.

It was written for the beta-test version of the Metacomco/Lattice C compiler, which doesn't like scanf to be used with structures. I had to use gets, and convert the string to an int, with the stcd\_i function. This bug will be fixed when the package is released.

```
    £include <fp1_stdio_h>
    £include <fp1_ctype_h>

struct P(
    char name[10];
    int age;
) person[100];

FILE *fp;

Main() [
    int i, nact;
    char temp[10], fname[10];

    Printf("FILE NAME: ");
    fflush(stdout);
    gets(fname);
    fp = fopen(fname, "w");
    if (fp == NULL)
        Error("FILE OPEN ERROR");
    For(i = 0 ; i < 5; I++){
        Printf("\nNAME: ");
        fflush(stdout);
        gets(person[i].name);
        Printf("\nAGE: ");
        fflush(stdout);
        gets(temp);
        stcd_i(temp, &person[i].age);
    }
    for(i = 0; i < 5; I++){
        nact = fwrite(&person[i], sizeof(struct P), 1, fp);
        if (nact != 1)
            Error("FWRITE ERROR");
    }
}

error(s)
Char *s; [
    puts(s);
    exit(0);
]
Leon Heller
```



Microcomputer Peripherals  
and Software

57 Repton Drive  
Haslington  
Crewe CW1 1SA  
Tel: (0270) 582301

COMPWARE QL PRODUCT PRICE LIST - AUGUST 1985

PRICES INCLUDE VAT AND DELIVERY

Hardware

Sinclair QL (Paion 2 packages)	£389-99
Microvitec CUB high res monitor with cable	£265-00
Swivel plynth for Microvitec CUB	£ 25-00
Sinlex 256K RAM expansion	£189-00 NEW LOW PRICE
2 slot expansion console (eg hold RAM+QDISC)	£ 65-00 NEW
4 slot expansion console (CST)	£199-99 NEW
CST Q-DISC QL floppy disc interface	£139-00 NEW LOW PRICE
3.5" or 5.25" single 360k disc system	£299-00 *
3.5" or 5.25" dual 360k disc system	£449-00 * & SEE BELOW
3.5" or 5.25" single 720k disc system	£329-00 *
3.5" or 5.25" dual 720k disc system	£499-00 *
CP80 80 column dot matrix printer	£225-00
Kaga Taxan Letter Quality Matrix printer	£309-00
Smith Corona Daisywheel printer	£219-00
10 5.25" floppy discs in plastic box	£ 29-95
1 3.5" floppy disc (DS/DD)	£ 5-49 (10 for £49)
4 microdrives in plastic wallet (as with QL)	£ 8-99 LOW PRICE
QL serial cable	£ 15-00
QL Centronics interface	£ 38-00

\* All disc systems include drive(s), power supply, all cables, Q-DISC interface, and extensive typeset manual. Just plug in and go.  
& WE HAVE LIMITED STOCK AT £349 (5.25" 360k dual only) - act fast

Software Utilities

Computer One Assembler	£29	Sinclair/Tony Tebby Toolkit	£24
Computer One Monitor	£24	Sinclair Technical Guide	£14
Computer One Pascal	£39	Metacomco Assembler	£39
Computer One FORTH	£39	GRAPHIQL	£34
Computer One Typing Tutor	£24	Sprite Designer	£10

Software Games

QL Backgammon	£15	Hopper (Microdeal)	£14
Zkul Adventure	£18	Cuthbert in space (Microdeal)	£14
West Adventure	£18	Lands of Havoc (Microdeal)	£19
Paion Chess	£18	Star Guard + Galactic Invaders	£14
Paion Match Point	£18	Hungry Harry	£10

PRICES INCLUDE VAT AND DELIVERY

WRITE OR TELEPHONE FOR FURTHER DETAILS OR GENERAL HELP



Microcomputer Peripherals  
and Software

57 Repton Drive  
Haslington  
Crewe CW1 1SA  
Tel: (0270) 582301

Medic Datasystems QL Expansion (Discs/Memory/Modem)

The Medic expansion cartridge for the QL is without doubt the most cost effective way of enhancing your QL with extra memory and/or disc drives. Later on you can upgrade the basic system you start with to any level including full 512K memory, dual 1 Mbyte disc drives, full specification modem and 240K EPROM experimental card.

Initially, the Medic cartridge is available in the following configurations at the prices shown which include VAT and delivery within the UK. These prices include a special discount for members of QUANTA.

Memory configuration	1 disc drive	2 disc drives
-	£239	£379
64k	£289	£429
128k	£319	£459
256k	£339	£489
512k	£429	£575

Disc drives are 3.5" Mitsubishi, give 720K formatted file space and run extremely quietly. The whole system is powered from an integral power supply within the disc drive case, and which connects to the expansion cartridge plugged into the side of the QL through a standard 34 way ribbon cable. The disc interface software is written by Tony Tebby, famed author of QDOS - the QL operating system - and the driving software for most of the disc interfaces available for the QL.

Extras

All the above systems come with a parallel printer interface, built in to the expansion unit, and the following FREE software:

- M-DESK Macintosh type single key depression user interface, for instant program switching and utilities selection.
- M-BASE Turns Archive into a menu driven database with single key commands
- M-ACCNTS Fully integrated sales, purchase, nominal ledgers and stock control
- M-KEY Single key entry of user defined text in ANY program
- M-SPELL Spelling checker
- M-MERGE Personalised mailshot program
- M-SQUEEZE File compression program
- M-BOOT Sets up RAM DISCS in memory and then automatically loads pre-defined files and programs
- M-TRANSFER Microdrive to disc transfer routine

**M. P. C. Software.**

**Software.**

<b><u>Programme name.</u></b>	<b><u>Publisher.</u></b>	<b><u>Price.</u></b>
Assembler	Metacomco	£ 35
BCPL	Metacomco	£ 50
LisP	Metacomco	£ 50
Pascal	Metacomco	£ 75
C	Metacomco	£ 85
Zkul	Talent	£ 16
West	Talent	£ 16
GraphiQL	Talent	£ 32
Microdrive Doctor	Talent	£ 20
Chess	Psion	£ 16
MatchPoint	Psion	£ 13.50
Hyperdrive	English Software	£ 13.50
Startrek & 3D Maze & Life	MPC Software	£ 7
M-Paint	Medic	£ 28
Hopper	Microdeal	£ 13.50
Cuthbert in Space	Microdeal	£ 13.50
Lard of Havoc	Microdeal	£ 18.50
Night Nurse	Shadow Games	£ 11
Galactic Invaders & Space Guard	Shadow Games	£ 13
Area Radar Controller	Shadow Games	£ 11
Paint Master	Shadow Games	£ 13
Quazimodo	Shadow Games	£ 11
Space Paranoids	Shadow Games	£ 11
Tally 1	Quest	£105
Tally 2	Quest	£ 55
CPM on Microdrive	Quest,	£ 85
CPM on disc	Quest	£ 55
O/S Card	Quest	£ 45
QL Assembler	Sinclair	£ 31
QL Tool Kit	Sinclair	£ 19
QL Touch 'N' Go	Sinclair	£ 19
Cash Trader	Sinclair	£ 63

**Hardware**

Medic disk interface, single 3 ½" disc Drive &. Parallel interface.	£ 240
Microdrive cartridges (1)	£ 1.80
Microdrive cartridges (4)	£ 7
Memorex 3 ½"SSDD diskette each.	£ 6
QL Computer	£ 380

Postage on all orders over £5 is free. On orders under £5 add 50P.

Special Offer: One free copy of Startrek/3D Maze/Life with all orders over £2.

**M. P. C. Software,  
72, Julian Road.  
West Bridgford,  
Nottingham. NG2 5AN.  
Tel: (0602) 820106.**